Biotic Prediction

Building the Computational Technology Infrastructure for Public Health and Environmental Forecasting

Software Requirements Document

BP-SRD-1.6

Task Agreement: GSFC-CT-1

November 30, 2003

1 Overview

1.1 Introduction

This project will develop the high-performance, computational technology infrastructure needed to analyze the past, present, and future geospatial distributions of living components of Earth environments. This involves moving a suite of key predictive, geostatistical biological models into a scalable, cost-effective cluster computing framework; collecting and integrating diverse Earth observational datasets for input into these models; and deploying this functionality as a Web-based service. The web-based application will present options for applying these series of models to the available datasets yielding predictive result sets. The resulting infrastructure will be used in the ecological analysis and prediction of exotic species invasions. This new capability will be deployed at the USGS Midcontinent Ecological Science Center and extended to other scientific communities and organizations through the *USGS National Biological Information Infrastructure* program. Figure 1 presents the components of this framework, their relationship to and interaction with the Invasive Species Forecasting System.

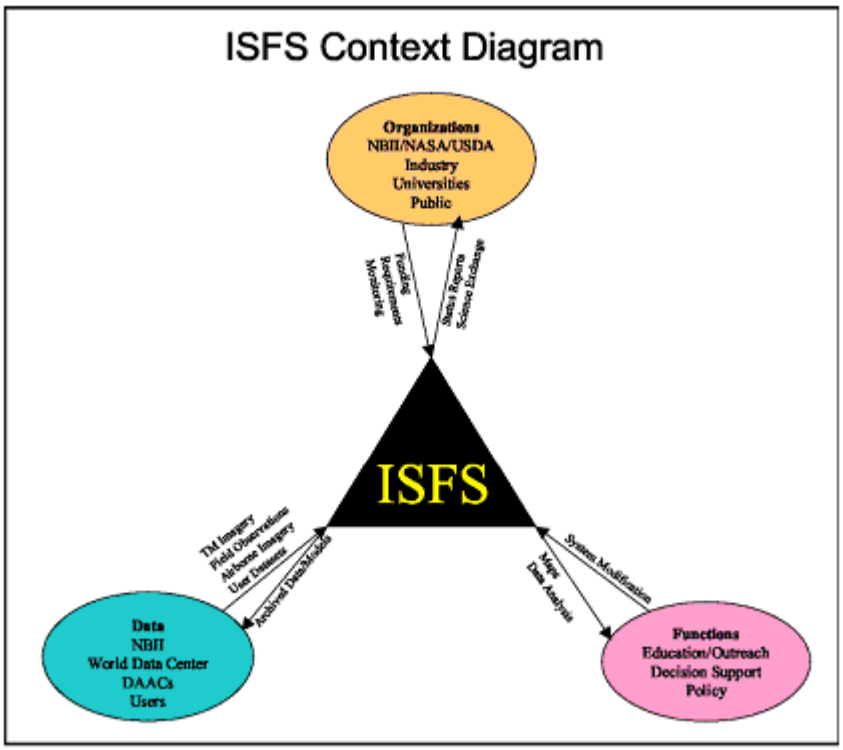

Figure 1 - Context Diagram

1.2 Document Versions

Date	Version	Description
Aug 22, 2003	1.4	Initial version for Milestone F
Nov 24, 2003	1.5	Revised submittal to CT relating to Milestone F. Applied all CT comments as noted in "Comments on Milestone F documents.doc"
Nov 30, 2003	1.6	Added Document Versions section Removed Requirements Trace Matrix into separate doc "BP-SRTM-1.0.doc"

Date	Version	Description
		Changed Functional Layout section to Conceptual System Layout

1.3 Referenced Documents

Document Title	Version	Date
Software Engineering / Development Plan (BP-SEP)	1.2	2002-09-26
Concept of Operations (BP-CONOP)	1.6	2002-10-17
Software Requirements Trace Matrix (BP-SRTM)	1.0	2003-11-30
Test Plan (BP-TP)	1.2	2003-12-03

1.4 Development Lifecycle

Development cycle milestones will be developed according to software engineering principles including analysis, design and requirements. Checkpoints between all phases of development will be scheduled to take place throughout the development cycle to assure all modules of the application are working well with each other as the development timeline moves forth. Checkpoints for the front-end, application, back-end and data layers of the project are necessary to avoid any potential delays in bringing all parts together as the end of the development cycle nears. The milestones and checkpoints will be rolled into the project schedule.

This document will incorporate new requirements that arise as part of the design, implementation & testing phases. The Software Requirements Document update & release cycle will tightly correlate with each development stage and be maintained according to standard software development lifecycle practices.

1.5 Document Overview

This document, the *Software Requirements Document,* enumerates the software requirements for the *Invasive Species Forecasting System* (ISFS). Some of these requirements are planned for immediate implementation in the software, and some are intended to be "design goals" for now and may be implemented in later builds of the software. By enumerating all of the requirements here, the software is described in its eventual, evolved state.

Section 2 provides a brief description of the system, its interfaces, and its functional subsystems. Sections 3 & 4 lay out the explicit list of Functional, Performance, and Documentation Requirements. Further revisions of this document will break down these requirements into more detail and add more specific derived requirements as needed. Section 5 provides an introduction and reference to the Requirements Trace Matrix.

2 System Overview

2.1 System Concept

The ISFS will provide an environment for the application of different statistical models to geolocated reference datasets. Users will be able to:

- Apply pre-defined models against existing datasets
- Add new or augment existing datasets
- Add new or alter existing models.

A primary goal of the evolved system is to optimize resource intensive computation by distributing it across a Linux cluster. This will minimize the time it takes to run the models and provide a better user experience.

In order to provide a baseline for comparison, the canonical system will:

- Show proof of concept for the canonical example and the validity of the example
- Establish metrics that can gauge performance of the system as a whole
- Provide an established baseline configuration from which future development can be compared
- Identify specific targets where computational processes can be enhanced to show scalar improvements in performance
- Identify the algorithms and tools that are used for processing and document these for the user.

2.1.1 Application Prototype

The web application which supports the above objectives is prototypical in design & implementation. It serves to allow canned models to be run for the purpose of satisfying project milestones, demonstrate use of the system, and identify requirements for the next phase of development. This next phase will extend the prototypical application into a production system.

2.2 System Environment

The *Invasive Species Forecasting System* will, for this stage of development be hosted on a Linux server at the NASA Goddard Space Flight Center, Building 28. It will use COTS tools including ENVI image processing, IDL, Fortran programs, and shell scripting languages. Datasets are already loaded and in the correct formats for processing and benchmarking the canonical example.

The establishment of a dedicated server will allow comparison of performance and product quality as different operating parameters are modified. Once the benchmarks for the baseline system have been recorded; the exact configuration of the dedicated server will be archived. Then, the server will be used to prototype different operational scenarios as documented in the BP-CONOP document.

Subsequent mods to the baseline will be scrupulously documented along with any changes in performance that are measured or observed.

2.3 Conceptual System Layout

The ISFS conceptually consists of 3 layers comprised of 7 functional elements as shown in Figure 2:

Front-End Layer:

1. GUI — A Graphical User Interface, where a web presence is maintained offering users universal access to the system.

Application Layer:

2. Ingest — The ingest subsystem will serve as the initial ”entry point” for all data used in the system, offering authorized users the capability of uploading data to be used by the system.
3. Pre-processing — Pre-processing activities merge ingested datasets to create a data product that can be analyzed in the modeling step.
4. Modeling Preparation — Application interfaces, allowing monitoring of and communication with Back End as well as information publication to the user.
5. Post-processing — The post-processing step applies the results of the modeling activities to generate products in the form of images and corresponding data for user analysis.

Backend Layer:

6. Archive — An archive subsystem will be coordinated by a database that will store pointers to archived files.
7. Compute Server – statistical algorithms run, thus predicting a certain species migration through or invasion of habitat based on remote sensing imagery and ancillary data layers.

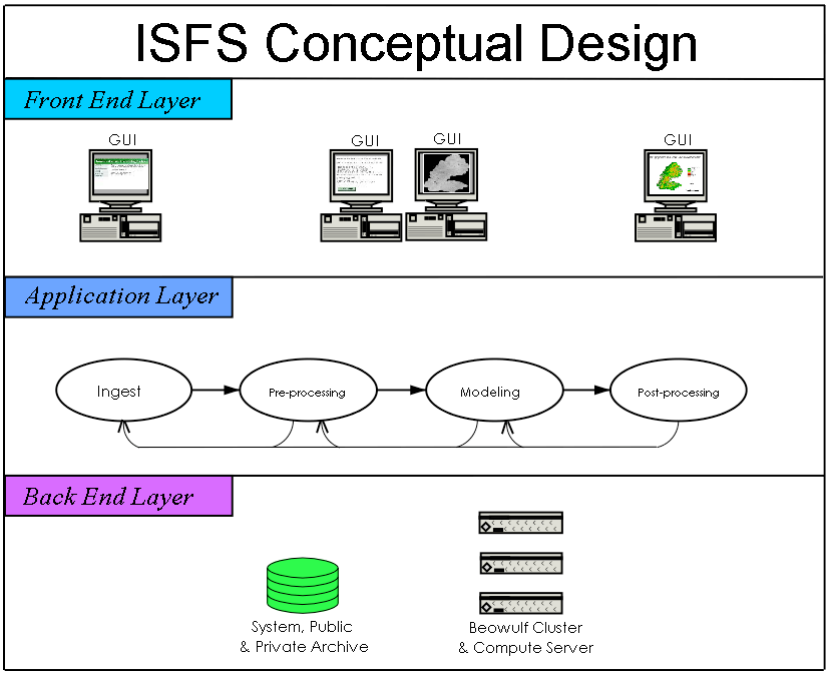

Figure 2 - ISFS Functional Layout

3 Functional Requirements

3.1 User Interface

3.1.1 Profile Database

3.1.1.1 The system shall maintain a profile database of users. The user profile will consist of the following persisted data: a) User's role, b) User's status , c) User's preferences

3.1.2 Roles

3.1.2.1 The system shall support various roles that control access to various capabilities of the system:

Administrator - This role provides complete access to the ISFS web system. An administrator would be able to manage other users and have access to their workspace, preferences and authorization information

Model Builder - This role is intended for the user that wishes to configure and tailor the model related tasks. The Model Builder is someone who has been authenticated and authorized to upload and ingest data in preparation for other users to perform repeated model runs. The Model Builder has registered with the system and maintains an active logon name and password to access the system. The Model Builder maintains a profile within the system that remembers the Builder's data selections and makes those selections available to the Builder upon login. The Model Builder does not have complete access to the ISFS web system, as complete access is the role of the administrator

Model User - This role is intended for a user to select specific input data and run it through a previously configured ISFS model run. The Model User has registered with the system and maintains an active logon name and password to access the system. The Model User maintains a profile within the system that remembers the User's data selections and makes those selections available to the User upon login. The Model User is someone who has not been authorized to upload and ingest data to the system

3.1.3 Graphical User Interface

3.1.3.1 The system shall include a Graphical User Interface ("GUI") to support user interaction with the system.

3.1.3.2 The GUI shall dynamically construct personalized web pages based on the Profile Database and the User's Role.

3.1.3.3 The GUI shall display predictive map and uncertainty map output.

3.1.3.4 The GUI shall invite all system users to register in order to use the system based on their roles.

3.1.3.5 The system shall allow the model builder to create new models w/in the Application functional layer.

3.1.3.6 The system shall allow the model user to select from an assortment of modeling techniques and to modify model parameters.

3.1.4 File Management

3.1.4.1 The user shall have the option of saving run results with annotations in personal repository.

3.2 Ingest

3.2.1 Validation

3.2.1.1 The system shall verify integrity, but is not required to validate data quality before ingest.

3.2.1.2 The system will accept data from an authoritative source. An authoritative source is someone who has registered with the system and been authorized to upload data to the system.

3.2.1.3 The system shall log all data sources.

3.2.2 Field Data

3.2.2.1 The system shall provide standard templates for ingesting field data in a tabular form.

3.2.2.2 The templates shall include all required fields to be captured.

3.2.2.3 The templates shall be in an accessible format (such as spreadsheet, database, or simple ASCII list). The format will include tab and white space delimited tables along with comma delimited.

3.2.3 Remote Sensing Data

3.2.3.1 The system shall support ingest from external satellite data archives (such as the Goddard DAAC).

3.2.3.2 The system shall support ingest of user-supplied satellite data or airborne imagery from digital files.

3.2.3.3 The system shall support ingest of user-supplied data, also known as user-supplied layers.

3.2.3.4 The system shall support ingest of user-supplied data files for ancillary layers. Ancillary layers consist of descriptive comments and other ancillary data that are relevant to the user-supplied data.

3.2.4 Data Acquisition

3.2.4.1 The system shall support secured ftp-push for ingest of user supplied data.

3.2.4.2 The system shall support automated secured ftp pull from external archives.

3.2.4.3 The system shall provide accounting and logging by requiring users' name and password.

3.2.4.4 The system shall maintain a list of external archives and required data sets and these shall be available to all users of the system.

3.2.4.5 The interface between the system and each external archive will be thoroughly documented as those connections are tested and verified.

3.2.5 Monitoring and Reporting

3.2.5.1 The ingest subsystem shall monitor the number and volume of data brought into the system and this information will be recorded.

3.2.5.2 The ingest subsystem shall produce data reports broken down by location, user and external archive.

3.2.5.3 The system shall generate and display associated metadata describing output files, runtime parameters, and performance statistics.

3.3 Pre-processing

3.3.1 Merge Data

3.3.1.1 The system shall merge ingested datasets into a data product that can be analyzed by the modeling subsystem.

3.3.1.2 The system shall perform re-sampling if the input data are not at the same resolution.

3.3.1.3 The data shall be converted to a common analysis format.

3.4 Modeling

3.4.1.1 The system shall provide the capability to specify response (dependent "Y variable") and explanatory (independent "X variable") variables from available ingested data.

3.4.1.2 The system shall provide graphical techniques (selection by rectangular or circular region) to explore the relationships between these variables. These can be thought of as "Zoom-in" and "add layers" functionality.

3.4.1.3 The system shall have the ability to "fit" models through Least Squares, or other optimization routines, such as Generalized Least Squares or Exhaustive Regression.

3.4.1.4 The system shall provide screening techniques to quantitatively assess which explanatory variables are related to the response variable (such as stepwise regression).

3.4.1.5 The system shall calculate geospatial statistics (such as variograms).

3.4.1.6 The system shall incorporate spatial structure into the modeling (such as generalized least squares or Kriging).

3.4.1.7 The system shall be able to output to the user model results and relevant model diagnostics.

3.4.1.8 The system shall be able to create new models by processing ingested datasets through the modeling subsystem.

3.4.1.9 The system shall allow the user to designate model techniques (e.g. stepwise regression) and parameters (e.g. nearest neighbor value for Kriging).

3.5 Post-processing

3.5.1 Re-projecting Data

3.5.1.1 The system shall display an image of the output data.

3.5.1.2 The system shall produce a data file suitable for re-projection by external COTS utilities.

3.5.2 Data Overlay

3.5.2.1 The system shall overlay output data with other layers as requested.

3.5.3 Metadata

3.5.3.1 Output data shall be packaged with appropriate metadata.

3.5.3.2 Each output data set shall be assigned a unique identifier.

3.6 Archive

3.6.1 Database

3.6.1.1 The Archive shall have a database that will store pointers to the archived files.

3.6.1.2 The database shall be able to refer to "internal" (stored in a local **File Store**) and "external" files.

3.6.1.3 All files ("internal" and "external") shall be indexed with a unique file ID.

3.6.2 Internal File Store

3.6.2.1 Files shall be stored in a logically arranged directory structure.

3.6.3 External Files

3.6.3.1 For external files, the archive system shall store a pointer or URL that can be used to retrieve and stage the files for subsequent processing.

4 Performance Requirements

4.1 CT Project Scaling Milestones

4.1.1 Improve implementation of PlantDiversity to:

4.1.1.1 Deliver canonical products 25X faster than the baseline implementation.

4.1.1.2 Accommodate 10X more sample data at 25X the time required in the baseline implementation and 10X larger area at 2.5X the time required in the baseline implementation.

4.1.1.3 Achieve the advanced application goals of delivering the canonical products 200X faster than the baseline implementation on a 256-node cluster and accommodating 100X more sample data 1000X faster than the baseline and 100X larger area 10X faster than the baseline on a 1024-node cluster.

4.2 Security & Reliability

4.2.1 User & sub-system security

4.2.1.1 [Refer to section 3.1.2.1 for functionality & privileges provided each user role.]

4.2.1.2 The chosen transport protocol for the application subsystems to interface with the cluster must be secure and extensible.

4.2.1.3 The application will be multi-user safe

4.2.1.4 Evaluate whether a user account and user-specific stored data expire

4.2.1.5 A ‘Guest’ account must be provided, allowing model-user privileges but no archival capability.

4.2.1.6 Appropriate user account standards must be defined & programmatically enforced. These are to include id & password format requirements, password encryption, account annotative information and account management.

4.2.2 Resource Utilization

4.2.2.1 There needs to be a data clean-up scheme that specifies what data is to be discarded, what data is to be kept, and for how long.

4.2.2.2 Internal storage requirements, including description of arrays, their size, their data capacity in all processing modes, and implied limitations of processing need to be determined & enforcement measures designed into the system.

4.2.2.3 The system shall include utilities to monitor node status, utilization statistics, communication, etc. for troubleshooting and analyzing system performance.

4.2.2.4 The system shall monitor and manage CPU utilization by each user as needed.

4.2.3 Stability & Maintenance

4.2.3.1 Exception handling will allow users sessions to recover gracefully

4.2.3.2 The model algorithms, originally developed in an informal research program, must be formally integrated into a production system.

4.2.3.3 Pre-processing needs to be well defined and developed.

4.2.3.4 The need for a job controller to manage model runs on the cluster will be evaluated. Cluster management tools to work with the job controller and ISFS need to be well defined

4.2.3.5 The system shall provide the ability for model runs to be gracefully paused, resumed or shutdown in the middle of the run.

4.2.3.6 The system shall support Linux. No designs to be portable to other platforms.

5 Requirements Traceability

5.1 Trace Matrix

This chart, found in the Software Requirements Trace Matrix (BP-SRTM), serves to provide a central source to track requirements through the development lifecycle. Information related to the source, dependencies, release, delivery and testing methods will be presented in this table.

5.2 Milestone F Requirements Summary

ID	Requirement
3.2.1.1	The system shall verify integrity, but is not required to validate data quality before ingest
3.2.3.3	The system shall support ingest of user-supplied data, also known as user-supplied layers
3.4.1.3	The system shall have the ability to "fit" models through Least Squares, or other optimization routines, such as Generalized Least Squares or Exhaustive Regression
3.4.1.5	The system shall calculate geospatial statistics (such as variograms)
3.4.1.7	The system shall be able to output to the user model results and relevant model diagnostics.
3.5.1.1	The system shall display an image of the output data.
3.5.1.2	The system shall produce a data file suitable for re-projection by external COTS utilities.
3.5.3.1	Output data shall be packaged with appropriate metadata.
3.5.3.2	Each output data set shall be assigned a unique identifier.
4.1.1.1	Deliver canonical products 25X faster than the baseline implementation.
4.2.1.2	The chosen transport protocol for the application subsystems to interface with the cluster must be secure and extensible.
4.2.3.6	The system shall support Linux. No designs to be portable to other platforms

Glossary

BP Biotic Prediction project
CT Computational Technologies project
CONOP Concept of Operations
COTS Commercial Off The Shelf
CSU Colorado State University
ESTO Earth Science Technology Office
GSFC Goddard Space Flight Center
GUI Graphical User Interface
ISFS Invasive Species Forecasting System
NREL Natural Resources Ecology Laboratory
SEP Software Engineering / Development Plan
SRTM Software Requirements Trace Matrix
URL Uniform Resource Locator